An Interop Publication

ConneXions®

The Interoperability Report

February 1993 Volume 7, No. 2

ConneXions —
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP.

E-mail: connexions@interop.com.

ISSN 0894–5926

From the Editor

Historically, the world of data networking has been divided into two distinct "styles": connection-oriented and connectionless. In a connection-oriented network, you open an end-to-end connection (a "virtual circuit") before transferring any data to the destination. All subsequent data packets travel over that same "pipe" in an orderly fashion, and underlying protocols ensure that the data isn't corrupted. Once all data has arrived safely, you close the connection. In a connectionless network, you send each individual packet on its way and let the network "do the rest." Each packet may travel a different route, some packets may be lost or duplicated, and packets may arrive out of sequence at the destination. It is the responsibility of the transport or application layer to "undo the damage" and ensure end-to-end reliable delivery. Most Local Area Networks (LANs) such as Ethernet are connectionless in nature, whereas most wide-area networks, such as those offered by telecommunications carriers are connection-oriented in nature. The best known example of the latter is X.25, the international standard promulgated by the CCITT.

In this issue we look at how X.25 can be used to interconnect LANs. The focus is on LANs using the TCP/IP protocol suite connected via routers, however, LANs supporting other protocol suites, multi-protocol routing and bridging are also discussed. The article is by one of Interop's instructors, Gil Falk of BBN Communications.

As networks grow larger and more complex the need for flexible configuration and management tools becomes critical. This is especially true when one has to deal with issues such as "acceptable use" and "policy routing." Our case study is written by Andy Adams and describes the new Merit Policy-Routing Configuration System that is used to manage the NSFNET T3 backbone.

Transmitting and receiving data at gigabit speeds is not the hardest problem related to the design of very high speed networks. A more difficult problem appears to be achieving gigabit speeds while also providing *service guarantees* such as bounding the maximum delay or ensuring that a video transmission will get enough bandwidth. Craig Partridge explains how such service guarantees can be provided using the concept of *traffic shaping*.

The debate over OSI has certainly not disappeared from the world of networking, nor should one expect it to vanish from the pages of *ConneXions*. Another double installment can be found in this month's "Letters to the Editor" section.

INTEROP 93 Spring is only a month away. Call 1-800-INTEROP or 1-415-941-3399 today for more information. We look forward to seeing you in Washington, DC in a few weeks.

LAN Interconnection Over X.25

by Gilbert Falk, BBN Communications

Introduction

Since the mid-1980s, the international standard X.25 has provided the basis for mission-critical multi-vendor wide area data networks (WANs). More recently, the need to interconnect a proliferation of local area networks (LANs) has led to the development of separate data communication systems consisting of trunk-connected bridges and routers. Given the worldwide installed base of robust X.25 WANs, it is important to develop a strategy for using these existing systems to support LAN interconnection. Although there are certainly situations where application performance requirements may preclude the use of an X.25 WAN, the "industrial strength" nature of X.25 networking has much to recommend it as a cost-effective substrate for LAN interconnection. There is also considerable experience which demonstrates that this type of system can support operational application requirements.

The objective of this article is to review the key networking issues that must be understood in order to successfully interconnect LANs over X.25. The focus is on LANs using the TCP/IP protocol suite connected via routers, however, LANs supporting other protocol suites, multiprotocol routing and bridging will also be discussed. Figure 1 illustrates the configuration that will be the basis for most of the discussion. A router (X.25 DTE) at each of a number of sites with one or more LANs is connected to a backbone packet switch (X.25 DCE) via an X.25 link. Hosts (computers, workstations, and PCs) may either be LAN-attached or connected directly to the X.25 backbone.

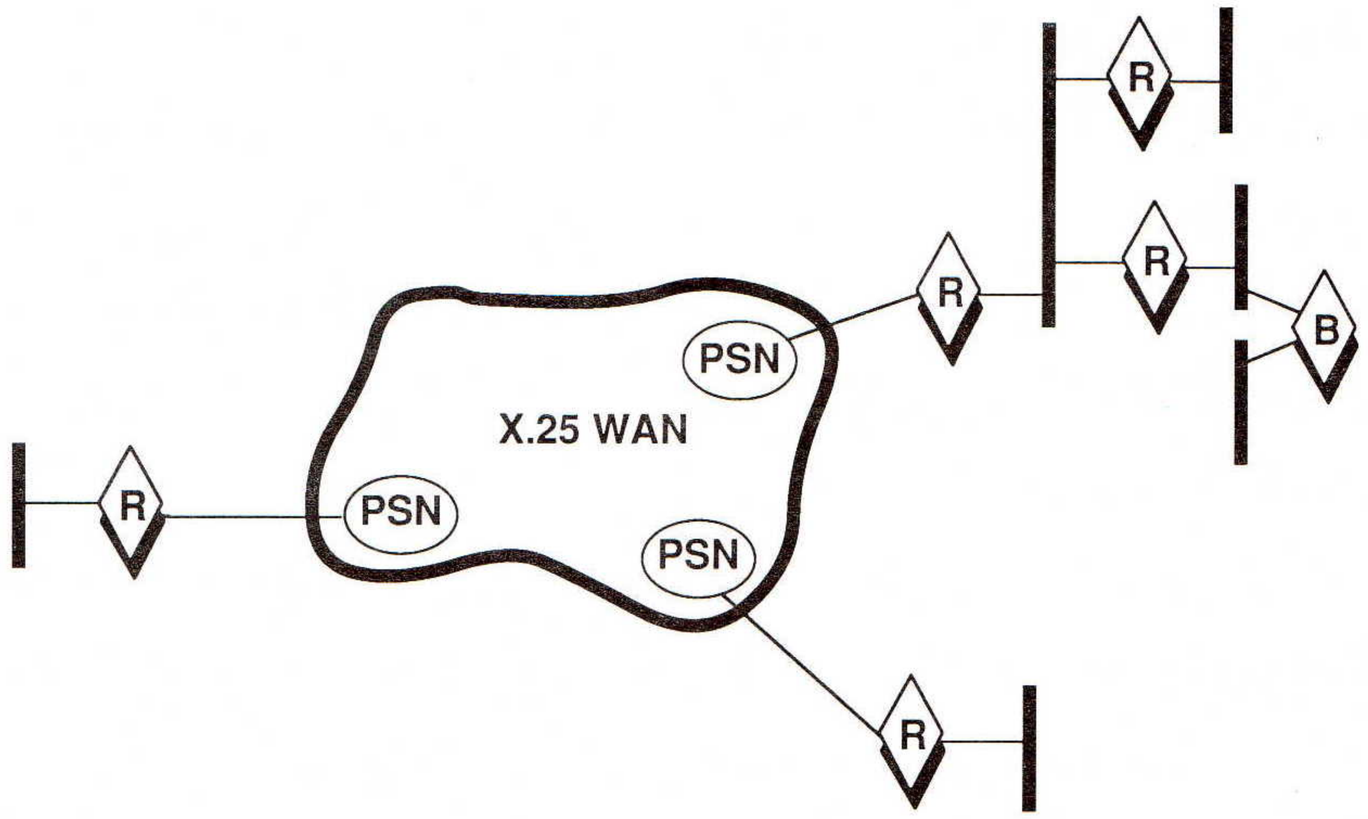

Figure 1: Basic Configuration

Connection management

X.25 networks provide communication between DTEs over both *Switched Virtual Circuits* (SVCs) and *Permanent Virtual Circuits* (PVCs). SVCs, established by the routers, are most commonly used to support LAN interconnection. There are two alternative strategies by which routers establish SVCs. With static SVCs, the routers establish virtual circuits among themselves at system start up time (either from a configuration file or via manual operator input). This fixed set of SVCs is used independent of the actual data flows that develop over time among the routers.

In contrast to this static SVC strategy, a dynamic SVC strategy establishes virtual circuits based on the actual demand for communication among the routers. This approach is the most common for supporting IP over X.25 and is available from most of the major independent router vendors including Cisco, Wellfleet and BBN.

It has the advantage of being suitable for efficiently supporting much larger LAN interconnections. Standards for dynamic SVCs are described in RFC 877 [3] which is likely to be replaced by [4] in the near future.

Dynamic SVCs operate in the following manner. When a router needs to forward an IP packet to another router (or an end system on the X.25 network), it first checks to see if a virtual circuit to that router (or end system) is already established. If the SVC exists, the router simply uses it. If no such SVC exists, the sending router first establishes an SVC and then forwards the packet over it. To establish the SVC, the router needs to map the IP address of the next router or end system into its corresponding X.121 address on the X.25 network. The next section describes three techniques for performing this IP-to-X.121 mapping.

Idle timers

Both routers and packet switches have limitations on the number of simultaneous virtual circuits that they can support. As a consequence, the implementation of dynamic SVCs must include a technique for managing this fixed SVC pool. One common SVC management technique is based on the use of idle timers. A parameter in the router is set to indicate that an SVC should be cleared after *N* seconds without any traffic. The value *N* must take into account the need to handle routing update traffic as well as data traffic if dynamic routing protocols such as RIP, IGRP, or OSPF are used. In particular, the SVC timeout should generally be set so that the SVCs do not need to be reopened for every routing update message if the line is not carrying any data traffic. Moreover, since packet switches may implement their own idle timers on SVCs, it is important to make sure packet switch and router idle timers are set in a consistent fashion. Typically one will want the packet switch SVC timeout to be greater than the router SVC timeout.

Another SVC management mechanism is the ability to (temporarily) preempt SVCs even if they are carrying user traffic. In general, routers implement various strategies for deciding which of the active SVCs should be closed. The goal of these SVC management strategies is to avoid excessive call setup and teardown ("thrashing") caused by oversubscription of the SVC pool. Thrashing will result in increased WAN traffic, node (router and PSN) utilization, and end user delay. Thrashing in a Public Data Network environment may significantly increase usage costs depending on the tariff structure. In practice, proper network engineering can eliminate these problems.

Address translation

Router-to-router communication involves the mapping of IP addresses into addresses understood by the network connecting the routers. Comer discusses the general issue of address mapping in [1]. As described in the previous section, X.25 call establishment requires that IP addresses be mapped into X.121 addresses. This IP-to-X.121 mapping provides a similar function to the *Address Resolution Protocol* (ARP) [1] which resolves IP addresses in an Ethernet environment. Since broadcast is not possible over X.25 WANs, some other technique needs to be employed.

Static tables

Two techniques are in common use for resolving IP addresses in an X.25 environment: static tables and functional mapping. The most general technique is the use of static mapping tables. This technique can be used for any private or public X.25 network. Each entry in the static mapping table contains a 32-bit IP address and the corresponding 14-decimal digit X.121 address.

continued on next page

LAN Interconnection Over X.25 *(continued)*

The router uses this table at SVC call setup time to establish the destination address of the next router. The shortcoming of this approach is that the mapping tables maintained in every router must be manually configured. For small networks, this is not a problem. As the size of the network increases, however, the problem of maintaining a consistent mapping database at every router location becomes more difficult.

IP to X.121 mapping

An alternative approach used in the *Defense Data Network* (DDN) [5] is based on a simple functional mapping between IP addresses and the corresponding X.121 address. Internet addresses for hosts on the DDN have the form <N.H.L.I> where N, H, L and I are octets of the 32-bit IP address and:

- N (network) = 26 corresponding to the MILNET portion of DDN
- H host) = number < 64 specifying a physical packet switch port
- L (logical address) = generally 0
- I (packet switch id) = number < 253 specifying a particular packet switch

The corresponding X.121 address is derived as 00000,III,HH,0000.

The fact that DDN manages its assignment of X.121 addresses makes this mapping mechanism a possibility. This approach may also be applicable in private X.25 networks where the switching nodes support a similar functional mapping (e.g., any network based on BBN's packet switches). However, for PDNs where there is no simple relation between the IP address and the X.121 address, one needs to rely on the static table approach.

Use of DNS

A third approach to providing IP-to-X.121 mappings is based on the use of a directory service. A proposal to expand the *Domain Name System* (DNS) to include such a capability for supporting IP over X.25 and ISDN networks can be found in [6]. To date, this approach has received little attention by system implementors.

Logical addressing

Because internetworks are continually evolving structures, it is important that address management mechanisms anticipate and facilitate change. To illustrate this point, consider the situation where the network manager needs to move a router and its attached LANs from one of the packet switches in Figure 1 to another packet switch. Such a reconfiguration might be carried out because some end users actually moved from one site to another or in order to better balance the load on the underlying transmission facilities. If the X.25 network supports only physical addressing, each network port will correspond to a distinct internet address and the change will require (1) modifying the configuration data in the relocated router to indicate its new internet address and (2) updating the routing tables in all the other routers. Step (2) can be a tedious and quite error-prone process if static routing tables are being used. If the X.25 network supports logical addressing, on the other hand, the move can be made with minimal change to the routers. The relocated router maintains a constant internet address which gets mapped into an X.121 "logical address" via one of the three techniques described earlier in this section. This logical address (not to be confused with the logical address portion of the IP address mentioned earlier) is mapped at X.25 call setup time into the corresponding physical address based on tables maintained within the packet switches.

For BBN packet switches, the process of updating logical addressing tables within the packet switches from the network management system is automated and operationally much easier than modifying static routing tables in each of the internet routers. Moreover, the use of logical addressing may obviate the need for IP routing exchanges across the X.25 backbone. In the case of one X.25 network with low speed trunks that BBN installed, this reduction in router control traffic overhead was particularly important in order to provide adequate capacity for user data traffic.

Routing

Closely related to the issues of connection management and address mapping is the issue of routing table maintenance. Each router maintains a routing table which maps IP destination address into an IP next hop address, i.e. the IP address of the next router along the path to the ultimate destination. It is this IP next hop address that is mapped into X.121 by one of the techniques described in the previous section.

Routing tables can be either static or dynamic. Static routing tables are conceptually very simple but must be manually configured and have the same disadvantage identified in the previous section for static address mapping tables. That is, they are difficult to manage for large networks. Dynamic routing tables, on the other hand, are automatically generated and maintained based on running one of the *Interior Gateway Protocol* (IGP) network routing protocols such as RIP, IGRP, OSPF or IS–IS. The advantages of the dynamic approach include simplified configuration and reconfiguration and support for automatic recovery from link or node failures. The primary disadvantage of running dynamic routing protocols in an X.25 environment is the traffic associated with routing updates and the potential requirement for SVCs between all router pairs to support this update traffic. As indicated earlier, in an environment with hundreds or thousands of X.25-attached routers and a limited SVC pool, the network designer needs to take steps to ensure that thrashing does not occur.

Solutions

There are a number of ways that a network can be engineered to avoid SVC thrashing. One approach is to simply guarantee that there are adequate resources to support IGP control traffic as well as data traffic. If this is impractical, the network can be structured as a logical hierarchy so that every router does not need to communicate its control traffic to every other router. This will reduce the requirement for SVCs on each of the connected routers. Hierarchical approaches are also commonly used in the case of large trunk-connected router internetworks to reduce routing overhead.

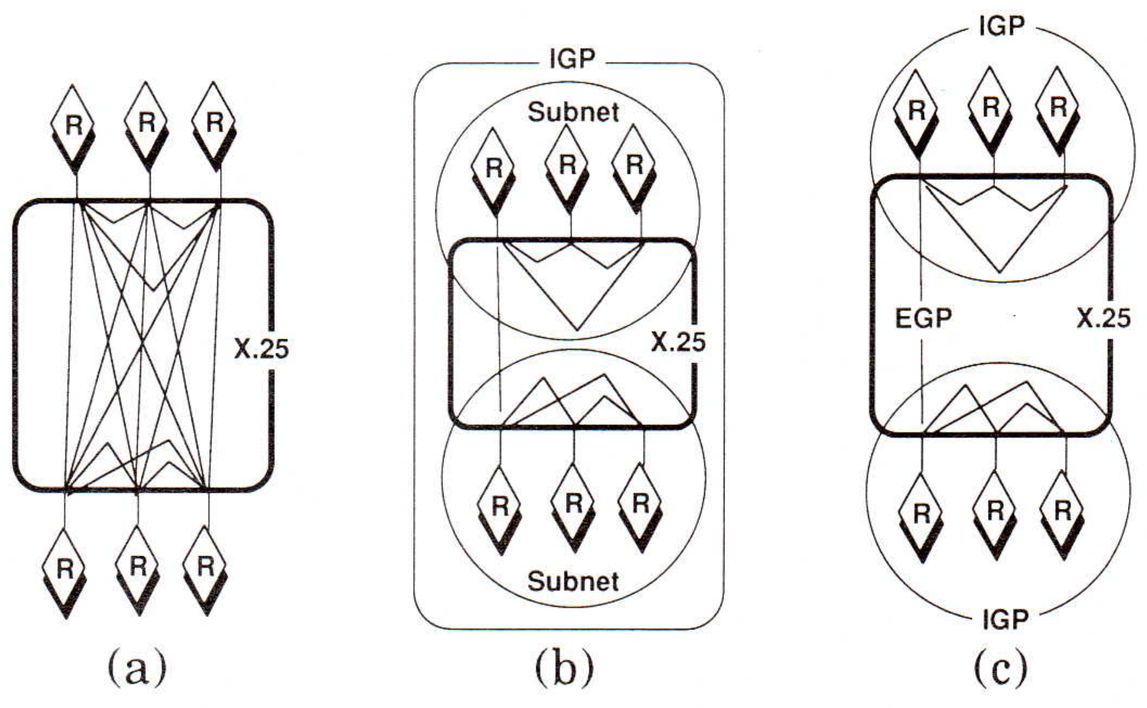

Figure 2: SVC Options to Support Dynamic Routing

continued on next page

LAN Interconnection Over X.25 *(continued)*

Figure 2 on the previous page contrasts a single level solution with two options for introducing a hierarchical structure. Figure 2a illustrates the single-level fully-connected solution with an SVC between every pair of routers on the X.25 backbone. Figure 2b illustrates the use of subnetting [1] to create the routing hierarchy within a single routing domain. Figure 2c illustrates the use of multiple independent routing domains to create the routing hierarchy. Routing information between independent routing domains is exchanged via some *exterior gateway protocol* (EGP) [1]. As illustrated in both Figures 2b and 2c, SVCs supporting routing updates only exist among routers within individual areas of a hierarchy and between specific routers connecting the areas. For the 6 routers illustrated in Figure 2, the hierarchical solution reduces the number of SVCs from 15 to 7. For very large internets, the reduction in the number of required SVCs can be much more dramatic.

Although the previous discussion has focused on the routing algorithm implemented by the routers, it should be understood that the packet switches independently execute their own routing algorithm in order to maintain SVCs in spite of link or packet switch outages. Historically, IGP implementations have been quite slow to recover from link outages, therefore, routers connected over X.25 substrates recovered much more quickly than those that were directly trunk-connected. With the increasing deployment of link state routing algorithms in routers (e.g., OSPF and IS-IS), however, one can expect that router and packet switch adaptation dynamics will become more similar.

Datagram fragmentation and reassembly

Datagrams are the network layer data units handled by routers. Datagram fragmentation and reassembly allows large datagrams to be carried independent of the *Maximum Transmission Unit* (MTU) of the underlying networks. IP datagrams larger than the MTU are fragmented by the routers and reassembled by the destination hosts. The IP protocol permits datagrams up to 65K bytes in length. Examples of common MTU values are 1500 bytes for Ethernet and 4K bytes for 4Mbps Token Ring.

In IP-over-X.25 environments, the maximum allowable datagram is also constrained by router buffering limits. The standard for encapsulating IP in X.25 requires routers to send each IP datagram as a "complete packet sequence" using the X.25 M-bit ("More data") in all but the last X.25 packet of the sequence. The router receiving the complete packet sequence must buffer all of the packets in the sequence and reassemble the IP datagram at the point that it exits the X.25 network. The proposed replacement for RFC 877 [4] specifies 1600 bytes as the minimum IP packet size that compliant routers must be prepared to reassemble from a complete packet sequence. This specification reflects current practice and permits maximum size Ethernet frames to be carried across an X.25 network without the need for IP fragmentation. IP datagrams that are larger than 1600 bytes (e.g., large token ring frames) must be fragmented by the routers as described above. In general, avoiding unnecessary IP fragmentation and reassembly is considered desirable [7].

Packet switches may also implement their own fragmentation and reassembly functions in order to permit pipelining of packet fragments and reduce the delay across the X.25 backbone. BBN's X.25 networks, for example, fragment X.25 packets at the source packet switch into as many as 8 packet fragments which are independently routed and reassembled at the destination packet switch.

If routers are connected across this type of X.25 backbone and the X.25 access links are relatively fast and error-free, the router and packet switch should be configured to use the maximum size X.25 packets that are supported by both the router and the packet switch. The packet network will generally be more efficient handling large packets and its internal fragmentation mechanism will ensure that transmission delay is also reduced. On the other hand, if the packet network does not implement internal packet fragmentation but forwards the entire X.25 packet as a unit between source and destination packet switches, the selection of X.25 packet size will be more critical. Although larger packets reduce the load associated with router reassembly of "complete packet sequences," larger packets also introduce additional transmission delay, especially if any of the trunks on the source to destination path are slow. The network engineer needs to understand the operation of the specific X.25 substrate as well as the user performance requirements before the LAN internet supported by that substrate can be properly configured.

Error recovery

In an internet consisting of trunk-connected routers, it is assumed that the underlying transmission links are high quality and have a low bit error rate. Routers do not implement error correction but rely on the end-to-end transport layer protocol (e.g., TCP) to detect and correct bit errors. X.25 networks, on the other hand, are designed under the assumption that the underlying transmission facilities may be noisy. They implement error control procedures both on the access links and on each of the links between the packet switches. In an environment where the links are nearly error free, the overhead associated with this additional functionality is unnecessary (and one of the justifications for the evolution of frame relay networks). On the other hand, if the network links are relatively noisy (e.g., some international circuits), then the link-by-link retransmission capability of X.25 networks can be of considerable benefit. Local retransmission of an X.25 packet or packet fragment by a packet switch can eliminate the need for retransmitting a great deal more data by the end user TCP module (all the data within the TCP window that has been sent prior to the receipt of the error notification). Local retransmission will also reduce the effective datagram delay in noisy communications environments.

Performance considerations

Network performance is one of the key considerations in deciding whether X.25 is suitable to support LAN interconnection. Network performance requirements are directly related to the specific applications which the network must support. Typical types of applications that one finds supported on interconnected LANs today include interactive host access, file transfer, electronic mail, file sharing, print service, database access, and graphic client-server interface. For some of these applications (e.g., interactive host access), the primary requirement is low delay. For other applications (e.g., file transfer), the primary requirement is high throughput. Some applications such as remote file sharing may require both high throughput and low delay. The delay, throughput and availability characteristics of X.25 networks in the context of supporting LAN-based applications are described in the following sections.

Delay

Network delay is an important performance requirement when a user is interacting with an application across a network. On-line transaction-oriented applications usually require that the average round-trip network delay be less than a few seconds. If the remote application system is echoing each character, delays of over 150–200 milliseconds are usually considered intolerable.

continued on next page

LAN Interconnection Over X.25 *(continued)*

There may also be delay requirements associated with supporting particular protocols. For example, DEC's LAT protocol which was designed for an extended LAN environment will time out connections where responses are not received within 80ms. Such application-specific requirements must be evaluated in each case to determine whether X.25 (or any other network technology such as Frame Relay) provides a suitable substrate for LAN interconnection.

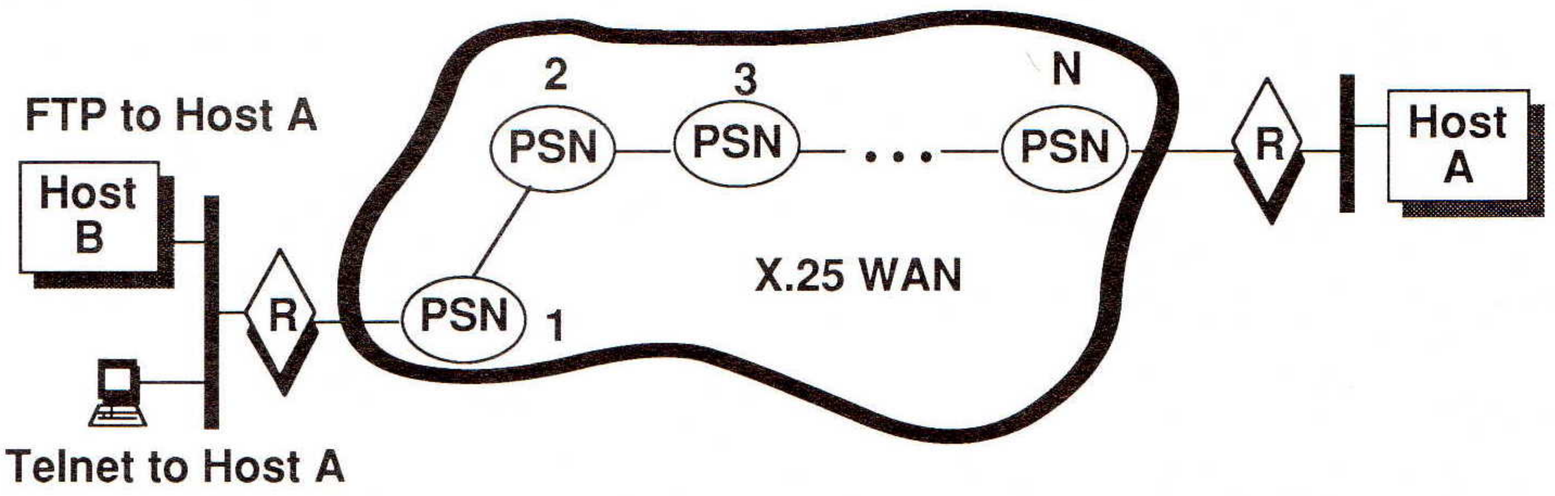

Figure 3: Network Model for Performance Analysis

Delay calculation

Figure 3 illustrates the elements that contribute to the delay within the X.25 environment. The path between routers consists of N packet switches and N+1 links. If all of the packet switches are identical and the links all run at the same speed, the average one-way delay for a single packet through this subsystem can be approximated by:

Delay = N x [Ave PSN Delay] + [N+1] x [Ave Link Delay] + Propagation Delay

where:

Average (Ave) PSN Delay depends on the performance characteristics and loading of the packet switches

Average (Ave) Link Delay depends on the size of the data unit being sent, the speed of the line, and the effective line utilization, and

Propagation Delay, the time that it takes for a bit to move across the sequence of transmission lines at the "speed of light," depends on the distance between the routers

Table 1 presents the result of applying this model to approximate the one-way transit delay for N = 3 as a function of link speed for remote echo and transaction-oriented transmissions under the following assumptions:

- Simple M/M/1 queuing models
- 50% utilization of resources (switches, trunks and access links)
- 5–7 ms Ave PSN Delay
- Propagation Delay = 35 ms
- Single character (remote echo) packet of 500 bits (including all overhead)
- Transaction-oriented application packet of 1000 bits (including all overhead)

Link Speed	*Remote Echo*	*Transaction*
9.6Kbps	467	883
64Kbps	113	175
1.544Mbps	59	61

Table 1: Average One-Way Transit Delay (ms)

From this table one can draw several conclusions. First, an X.25 network running with voice grade links will likely be adequate for supporting transaction-oriented traffic but will be inadequate to support remote character echoing. Increasing the link speed from voice grade to DS0 (64Kbps) brings the packet transit delay down to the point where remote character echoing is practical. A fast touch typist may have difficulty with a 226ms round trip delay, but most users will find 64Kbps operation acceptable. Finally, one can see that a high speed PSN network operating at T1 rates can provide low delay for transactions essentially independent of packet size and thus could easily support remote echo as well as local echo interactive applications for all users.

Throughput

Network throughput is the most important performance requirement when an application is transmitting a large volume of data across the network. The requirements for network throughput can vary widely. At one end of the spectrum remote file access (e.g., NFS) operations need to be completed within tens of milliseconds if the user is to find the remote file server an acceptable substitute for a local device. Neither an X.25 packet network nor any other widely-deployed WAN technology (e.g., Frame Relay) is suitable to support this type of operation. Remote file access works best at LAN speeds. On the other hand, acceptable times for file transfer and electronic mail operations are often measured in minutes. An X.25 backbone will provide a level of service that is more than adequate for these applications in most cases.

Assuming that the routers themselves are not the bottleneck, the throughput that can be sustained between a pair of routers over an X.25 virtual circuit is a function of a variety of factors. These factors, many of which are dependent on the characteristics and internal operation of the packet network, include:

- *Link speeds* (both trunk and access links): Voice-grade analog lines operating up to 19.2Kbps, digital links at 56 (or 64) Kbps, and T1 circuits are used, with 56 (64) Kbps being the most common.
- *Packet switch processing rates:* Raw packet switch processing rates of hundreds to thousands of packets per second means that this is probably not the limiting factor for a single application flow.
- *Competing traffic:* Although the overall traffic load does not constrain the maximum flow that a network can support, it will impact the actual throughput sustainable by each of a competing set of flows in an operational environment.
- *Congestion control limits:* Maximum link and packet switch utilization levels imposed by congestion management mechanisms, while essential for proper operation of the network at high load, do impose an ultimate limit on the throughput of individual connections.

continued on next page

LAN Interconnection Over X.25 *(continued)*

- *Data unit overhead:* including packet network and TCP/IP data unit headers and trailers on both the access circuits and the inter-switch trunks. TCP and IP each add 20 bytes of overhead to each X.25 packet independent of the underlying technology. Packet network overhead is vendor specific. Based on BBN's experience, assuming 1024 byte X.25 packets, a reasonable estimate for this component of overhead is 5% on access circuit and 15% on trunks.

- *Control traffic:* Control traffic includes both control messages that are a function of user traffic (e.g., internal acknowledgements) and control traffic related to background processes (e.g., trunk monitoring, routing updates and network management). In BBN packet networks, the traffic dependent component is typically negligible because of the ability to piggyback acknowledgements. Similarly, trunk monitoring traffic is also quite small. Routing update and network management traffic are dependent on the size of the network. In BBN's experience, network management traffic is typically less than 5% of the traffic on a trunk and dynamic adaptive routing update traffic is on the order of 2Kbps for a 250 packet switch network.

- *Maximum X.25 packet size and window size:* As is the case with any sliding window protocol, the combination of maximum packet size and maximum window size supported by the packet network bound the quantity of application data that can be sent in one X.25 network *round-trip time* (RTT). Assuming RTT = 300ms, the maximum throughput per SVC that can be supported with a maximum packet size of 1024 bytes and a window size of 7 is about 191 kilobits per second. Some systems only support much smaller maximum packet and window sizes. With a maximum packet size of 128 bytes and a window size of 2, the maximum throughput is limited to about 7 kilobits per second. Proper configuration of these two parameters is critical to proper network operation. It should be noted that some routers support multiple SVCs between a pair of IP addresses in order to support increased throughput.

- *Characteristics of the layer 3 protocol encapsulated in X.25:* It is important to note that some network layer LAN protocols may prove particularly troublesome in WAN environments. Novell's IPX protocol, for example, has traditionally limited both the size of the datagram that it sends (576 bytes per datagram) and the maximum number of outstanding datagrams (only 1 per acknowledgement). Recognizing its need to address this shortcoming as it tries to move from its position as a workgroup LAN supplier to an enterprise network supplier, Novell has recently announced enhancements to IPX which will support both larger datagram sizes and a greater number of outstanding datagrams. The characteristics of the layer 3 LAN protocol are important independent of whether the substrate is X.25 or any other WAN technology.

From these considerations it is possible to draw some conclusions about the maximum throughput that a LAN application is likely to achieve over an X.25 substrate. Assuming 64Kbps links, the primary limit will be the fraction of the 64Kbps capacity that is available. It is reasonable to expect that a single user can transfer data at 40–45 Kbps in a lightly loaded network.

As the load increases, the user will have to share this capacity with others and the maximum throughput per user (i.e., per virtual circuit) will decrease. If the packet network is configured with T1 trunks, much higher throughput can be achieved and packet and window size constraints are likely to dominate as described above. X.25 defines modulo 128 extended packet sequence numbering as an optional user facility that can be applied to eliminate this constraint. Alternatively, as indicated above, multiple virtual circuits can be established between a pair of routers to provide improved throughput.

Availability

As internetworks become more involved with supporting mission-critical applications, maintaining high end-to-end availability becomes increasingly important.

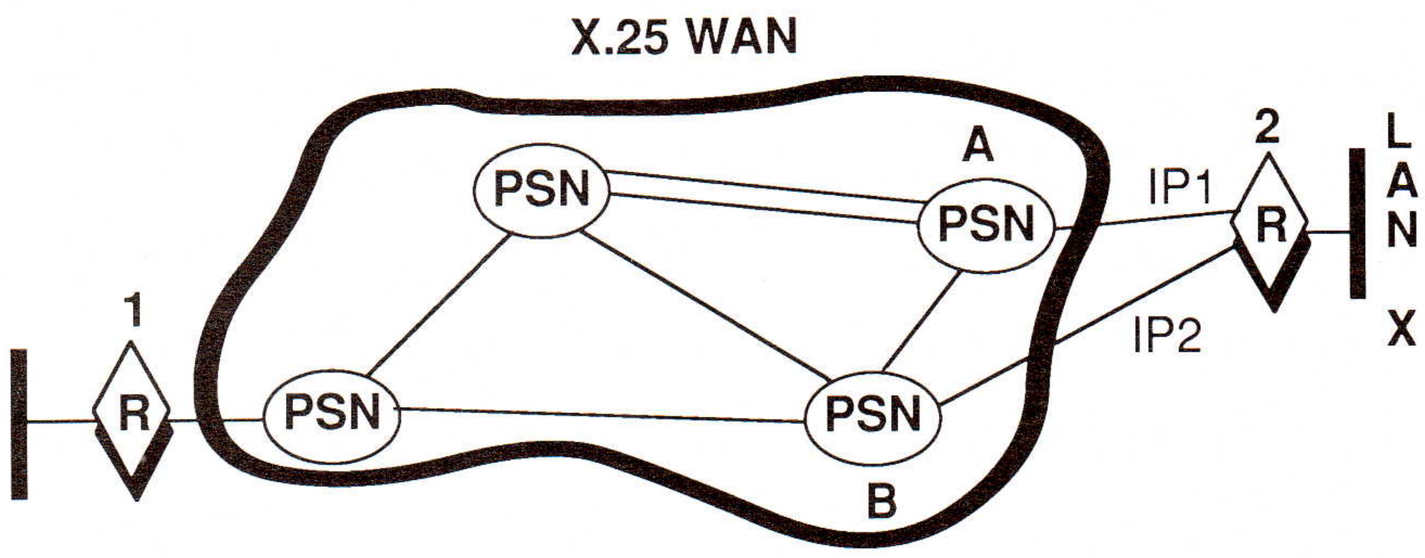

Figure 4: Features Supporting High Availability

Figure 4 illustrates several of the features of packet switching networks which help provide high availability. First, packet switches are generally connected in a mesh topology providing multiple independent paths between a source and destination. In addition, parallel trunks between the same packet switches can be configured to provide incremental capacity as well as redundancy. Packet switches continually monitor the status of interswitch trunks so that failures can be quickly identified. With BBN packet switches, dynamic adaptive packet-by-packet routing will identify alternate routes to rapidly recover from intermediate trunk and packet switch failures without loss of the X.25 virtual circuit.

Dual homing

To provide high availability on an end-to-end basis, one needs to protect against end node and access link outages as well as failure of trunks and intermediate nodes. One strategy is to provide dial backup between the router and a single packet switch incorporating redundant hardware. Dual homing, the ability to connect a router (or any host computer) to multiple packet switching nodes, offers another approach to providing protection against such outages. In Figure 4, router 1 can reach LAN X over two independent access paths via router 2. If the packet network supports logical addressing as described earlier, router 1 can send all datagrams destined for LAN X via address IP1 and the logical address mapping mechanism will automatically provide the switchover from PSN A to PSN B if PSN A or its access link to router 2 were to fail. If the packet network does not support logical addressing, the router can still understand that either address IP1 or IP2 can be used to reach LAN X. Moreover, since IP1 and IP2 are equidistant from router 1, router 1 may be able to be configured to perform load balancing between the two IP addresses sending alternating datagrams over a pair of virtual circuits. Although not illustrated in Figure 4, the router originating the traffic can also take advantage of the reliability provided by dual homing.

continued on next page

LAN Interconnection Over X.25 *(continued)*

Multiprotocol support

The options for handling multiple protocol suites in an X.25 environment are similar to those that are available with trunk-connected routers. One option is to route only IP and tunnel all other protocols across the IP network, that is, encapsulate network layer packets of other protocol suites within IP packets. While tunneling may be useful as a solution for a small number of protocol suites in a predominantly IP environment, it is not likely to be the best general solution for a large heterogeneous network. In such an environment, the approach taken today by most major router vendors (e.g., Cisco, Wellfleet, etc.) is to route each of the protocol suites independently.

Encapsulation

An important issue to be considered when handling protocol suites independently is the standardization of the encapsulation within X.25. In particular, although RFC 877 defines a standard for encapsulating IP within X.25, there are currently no Internet standards for encapsulating other protocol suites. While the OSI protocol specifications define a multiprotocol encapsulation standard, this standard is currently not widely implemented. In practice this means that routers from different vendors may interoperate over X.25 in an IP-only environment but are unlikely to interoperate over X.25 in a multiprotocol environment. Currently an organization must buy all of its routers from a single vendor to support multiprotocol operation. This situation should improve over time, however, since the proposed replacement for RFC 877 [4] specifies standards for multiprotocol encapsulation over X.25.

Bridging

Since not all network protocols are routable (e.g., LAT and NetBIOS), multiprotocol "routers" used to interconnect heterogeneous LANs are often required to support layer 2 bridging functions as well as layer 3 routing functions. Protocols designed to operate in a bridged building or campus environment often make implicit delay, bandwidth, error rate and functionality assumptions that may not be true in a WAN environment. The network designer must carefully evaluate the requirements of each supported protocol suite in the context of the underlying X.25 substrate. The strict delay requirement related to supporting the LAT protocol was previously noted. Other examples are the large bandwidth and multicast capability assumptions made by NetWare in conjunction with its *Service Advertisement Protocol* (SAP) and NetBIOS implementations in conjunction with name management and session establishment procedures.

Routers implementing bridging functions will typically run the same bridge control algorithms implemented by stand-alone bridges. In particular, routers supporting transparent bridging will run the spanning tree algorithm to select links used for packet forwarding and routers supporting source routing bridging will exchange explorer packets in order to discover routes to new destinations. The additional traffic associated with these control processes needs to be factored into the overall internet design. An advantage of running bridging over X.25, however, is the more reliable SVC connections between bridge modules provided by the backbone. Recovery from outages by the X.25 substrate can significantly reduce the need for slower and more inefficient recovery by the bridges themselves.

Congestion and priority management

Network congestion occurs when the aggregate demand on a subset of network resources exceeds their capacity. Since network congestion causes throughput to decline and delay to increase, the goal of the network provider is to ensure that network congestion does not occur or at least does not persist.

Mechanisms to address internet congestion management have been proposed, standardized, and implemented, however, this area is still in its infancy and most trunk-connected router internets currently avoid congestion by overconfiguring circuit and router capacity.

X.25 backbones, on the other hand, implement sophisticated congestion control algorithms permitting the network to operate at considerably higher link and node utilization levels. For example, BBN's family of packet switches implement a flow rate-based algorithm where information to support congestion control is explicitly piggybacked on routing messages, allowing source nodes to adjust their flows to each destination node. Routers connected over BBN X.25 backbones can consequently make much more cost-effective use of trunk circuits than routers that are directly connected over trunks.

Priority offers another strategy for managing the allocation of limited resources among competing traffic flows. Priority is supported in BBN X.25 networks as an extension to the CCITT X.25 standard. A priority is assigned to each SVC at call setup time. Packets associated with that SVC are handled at the selected priority level as they move across the X.25 backbone. This mechanism can be used to support "Type of Service" distinctions that are provided by higher layer protocols such as IP.

Conclusion

X.25 networks provide an attractive WAN substrate upon which to support a wide range of LAN-based applications. Many interactive terminal-to-host, file transfer, electronic mail, and client-server applications can and have been run successfully across X.25 networks. The worldwide installed base of X.25 networks, both public and private, provides an instantly available infrastructure for supporting LAN-to-LAN connectivity. In addition, X.25 interfaces are available for most routers and hosts that one might wish to connect to an X.25 WAN. Many of the functions of X.25 networks provide an "industrial strength" substrate that is needed to support mission-critical LAN-based applications. Congestion management, link state dynamic adaptive routing, and priority data handling, which have been operational in packet switches for years, are just beginning to be implemented by some router vendors. In international and military environments with noisy communications media, X.25's error correction facilities provide a valuable addition to end-to-end error control. Finally, although tools for managing trunk-connected routers are becoming increasingly available, X.25 WANs are still considerably more manageable, particularly in the areas of performance instrumentation and accounting management. It is our belief that X.25-based internets will continue to complement the emerging higher-performance but lower functionality network technologies for some time to come.

References

[1] Comer, D., *Internetworking with TCP/IP Volume 1,* Prentice Hall, Englewood Cliffs, N.J., 1991.

[2] Deasington, R. J., *X.25 Explained: Protocols for Packet Switching Networks,* 2nd ed., John Wiley & Sons, New York, 1984.

[3] Korb, J. T., "A Standard for the Transmission of IP Datagrams Over Public Data Networks," RFC 877, September 1983.

[4] Malis, A. G., Robinson, D., and Ullmann, R. L., "Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode," RFC 1356, August 1992.

continued on next page

LAN Interconnection Over X.25 *(continued)*

[5] Morales, L., and Hasse, P., "IP to X.121 Address Mapping for DDN," RFC 1236, June 1991.

[6] Everhart, C. F., Mamakos L. A., Ullmann, R., and Mockapetris, P.V., "New DNS RR Definitions," RFC 1183, October 1990.

[7] Kent, C. A. and Mogul, J. C. "Fragmentation Considered Harmful," Proceedings of SIGCOMM 1987, pp. 390–401.

[8] Jain, Raj., "Congestion Control in Computer Networks: Issues and Trends," *IEEE Network Magazine,* May 1990, pp. 24–30.

[9] Malis, A. "Multiprotocol Encapsulation over Frame Relay," *ConneXions,* Volume 6, No. 8, August 1992.

[10] Kozel, E., "The Cisco/DEC/NTI/StrataCom Frame Relay Specification," *ConneXions,* Volume 5, No. 3, March 1991.

[11] Vair, D., "Components of OSI: X.25—the Network, Data Link, and Physical Layers of the OSI Reference Model," *ConneXions,* Volume 4, No. 12, December 1990.

GILBERT FALK is director of strategic systems analysis at BBN Communications. Since 1973 he has worked with government and commercial clients to address their networking problems. Dr. Falk holds bachelor's and master's degrees in electrical engineering from MIT, and a Ph.D. in computer science from Stanford University. E-mail: `gfalk@bbn.com`.

Write to *ConneXions*!

Have a question about your subscription? Are you moving, and need to give us your new address? Suggestions for topics? Want to write an article? A letter to the Editor? Have a question for an author? Need a *ConneXions* binder? Want to enquire about back issues? (there are now 71 to choose from; ask for our free 1987–1992 index booklet). We want to hear from you. Contact us at:

ConneXions—The Interoperability Report
480 San Antonio Road, Suite 100
Mountain View, CA 94040–1219
USA
Phone: +1 415-941-3399 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-949-1779
E-mail: `connexions@interop.com`

The Merit Policy-Routing Configuration System

by Andy Adams, Merit Network, Inc.

Introduction

The NSFNET T3 backbone currently routes traffic among nearly 7,000 networks, each of which may have multiple backbone connections and which may, based on arbitrary policy restrictions, prefer some connections over others. Combine this with the fact that 1,400 new networks are connected to the backbone every six months and you have a unique network management challenge. A SPIRES database system constructed in 1988 currently maintains backbone policy-routing information. The system has proven not to be as flexible and scalable as we would have liked and thus we are developing its replacement. This article describes the new Merit Policy-Routing Configuration System being developed to meet this network management challenge.

Policy-based routing

In order for two networks to exchange data there must be at least one path between them and often multiple paths exist to improve reliability. The drawing below illustrates such a situation where there are two paths between networks X and Y, both of which include the backbone.

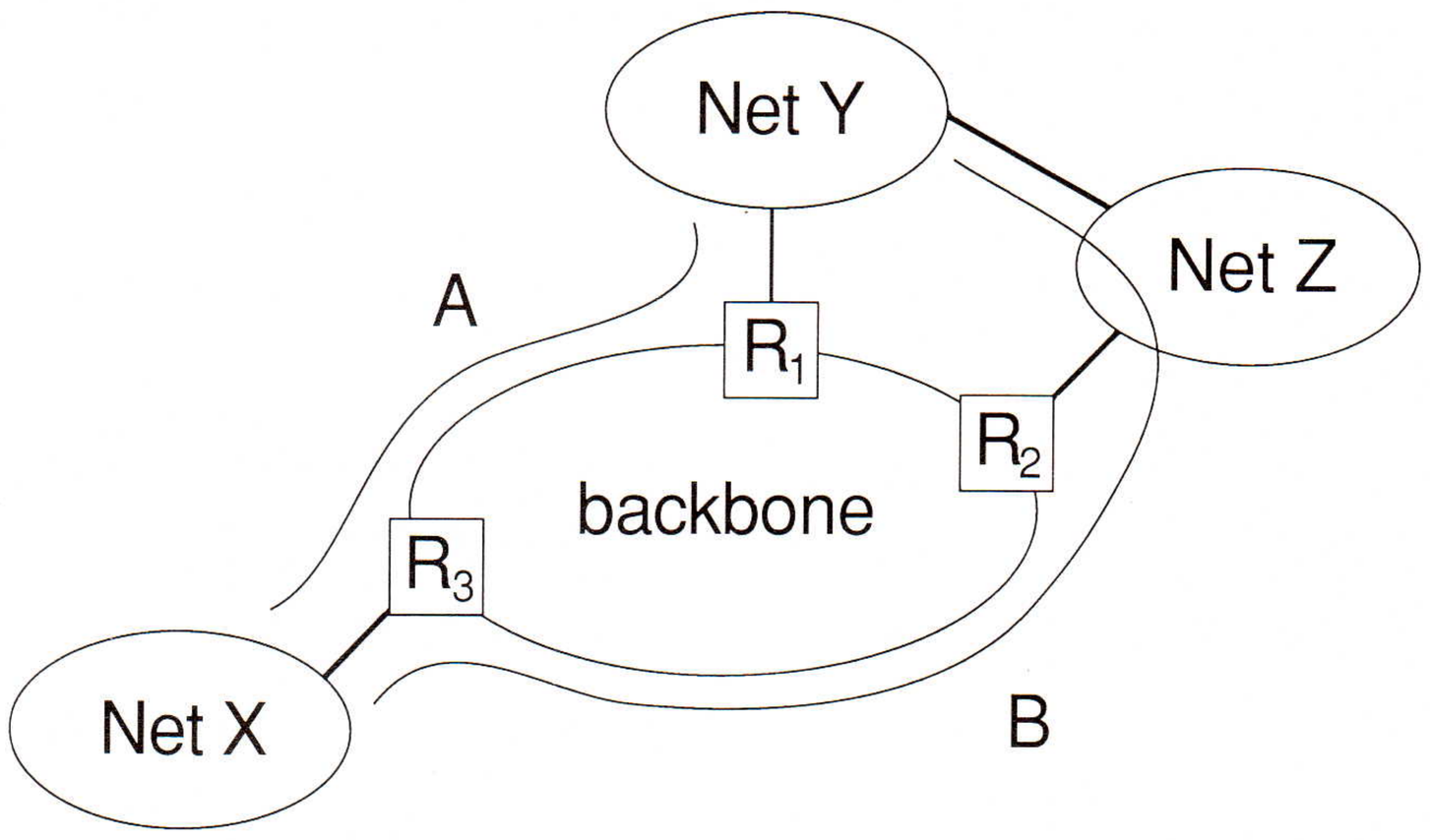

Figure 1: A Sample Peer Network Configuration

All paths are not created equal and a network's administrator may decide for any number of reasons that his traffic should traverse path B only if path A is unavailable. Reasons for this might stem from available bandwidth, distance as measured by number of routers traversed or a transit network's acceptable use policy. In the above example, the best network latencies between networks X and Y might be achieved by minimizing the number of hops between the two, and thus traffic from X destined for Y should prefer path A. In the event that path A is unavailable, the longer path through transit network Z should be taken. These arbitrary preferences placed on a network's external connections are called the network's routing policy.

Routing policy on the backbone is enforced with configuration files residing on each backbone border router. In this discussion a border router should be thought of simply as an entry or exit point from one network to its peers. Each configuration file is unique and describes which exit point traffic destined for a particular network should take.

continued on next page

Policy-Routing Configuration System *(continued)*

These files also describe on a per-network basis the preference placed on each exit point. In the above example, R1's configuration file would indicate that it is the primary exit point for traffic destined for network Y, while R2's would indicate that it is the secondary exit point.

System overview

The configuration system consists of the following components:

- *Configuration Requests:* An authorized requester must submit a configuration request before a network may pass traffic on the backbone. These requests, usually in the form of e-mail templates, are sent to a special configuration address and specify what paths to a particular network exist and what is the order in which these paths are preferred.

- *Configuration Operator:* A Configuration Operator is responsible for reading connection requests, collecting the appropriate authorizations from the engineering group and administrators of transit networks, entering the information into the database, and generating the required files. A Configuration Operator is also often responsible for working as a technical consultant with a peer network's engineers, helping to resolve various routing issues.

- *Configuration Operator's User Interface:* A Configuration Operator enters routing information into the database with the Configuration Operator's User Interface. This interface allows the operator to view the information contained in the database as the routing entities being modeled, instead of as tables or records.

- *Configuration Files:* Configuration files are those files placed on each backbone border router describing exactly which router is the exit for traffic destined for a particular network.

- *NOC Clients:* NOC clients are programs that query the database remotely for routing and topology information. They are used by NOC operators and network engineers to obtain information necessary for network trouble-shooting and planning.

- *Database:* The Policy-Routing database forms the core of the Merit Policy-Routing Configuration System, containing information about what networks are reachable via the backbone and from what points. It also contains information which, while not central to routing, is necessary for administering the backbone. This is information such as network contact names and phone numbers and network organization information.

- *Electronic Reports:* Several reports containing information about network connections and preferences are generated after each config run. These reports are made available via anonymous FTP and are used by some regional networks to configure their own routers.

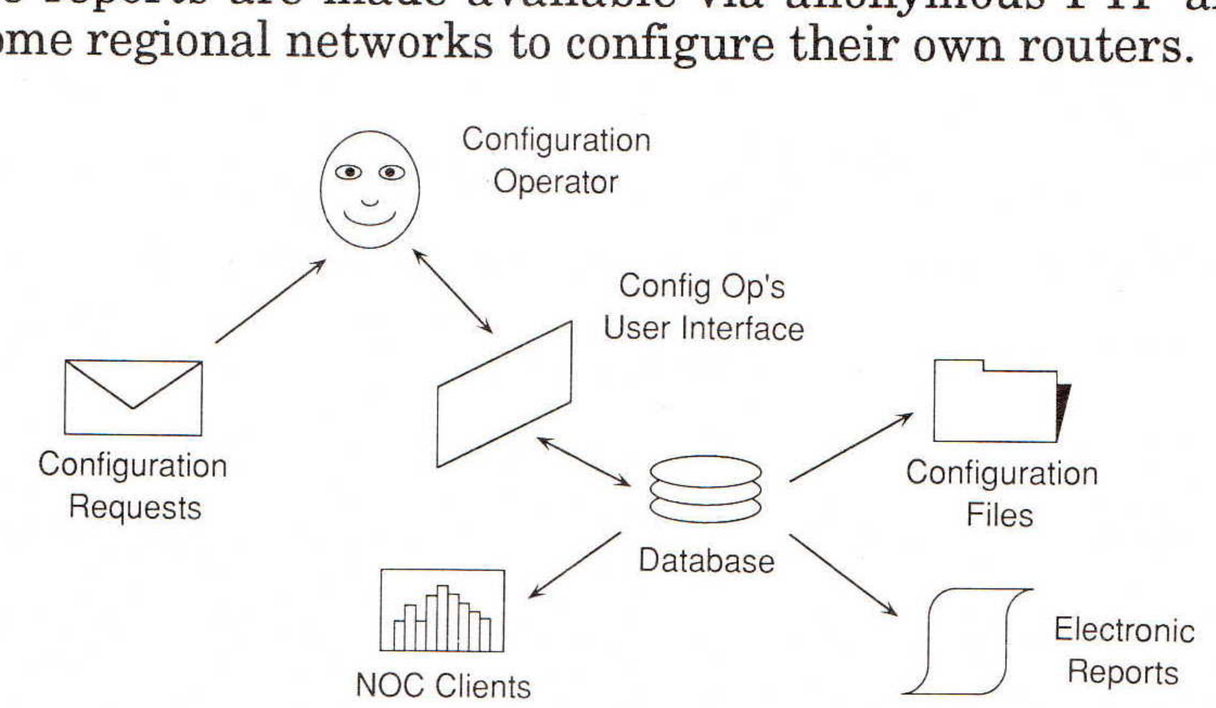

Figure 2: Configuration System Components

Example

To illustrate the configuration process, let's say that Grand Valley State University decides that it wants an Internet connection in order to access different super computers around the country. Reliability is important so they decide to obtain connections from two different backbone peer networks, A and B, who connect to the backbone at different points. Grand Valley also decides that peer network A will be their primary path and B will be their backup path. A representative from one peer network, let's say A, now sends an e-mail *Network Addition / Change Request* (NACR) to Merit asking that the backbone route traffic to Grand Valley's network first through peer A and second through peer B. A Configuration Operator reads this NACR and then sends a message off to peer B's representative asking if it is indeed OK for their network to act as a backup path for Grand Valley. When a confirmation is received, the operator enters the new network address, organization and path preferences into the database. She then generates two new configuration files, one for each point at which the peer networks are connected, and moves them to their respective routers; traffic may then be exchanged with Grand Valley. In addition to the configuration files, the operator also generates several reports reflecting this new backbone configuration and makes them publicly available on `nis.nsf.net`.

Design constraints

The overall design of the configuration system has been shaped by three fundamental constraints:

- The system must be flexible in order to keep pace with the constantly evolving routing architecture of the Internet. It must be able to accommodate new protocols and routing paradigms with a minimum amount of programming effort.

- The system must scale well in the face of Internet growth. It must be able to adequately handle the hundreds of new NSFNET connections that are approved each month without sacrificing performance or maintainability.

- The system's components should be independent from one another, allowing us to transparently change major components of the system with a minimum of programming effort. Examples of this would be changing the operating system or the database engine.

Layered architecture

The above design constraints have led us to build a system with a layered architecture. The layers are, from the bottom up, the Operating System, Database Engine and Schema, Database Remote Procedure Call, Validation, Gizmo and Application.

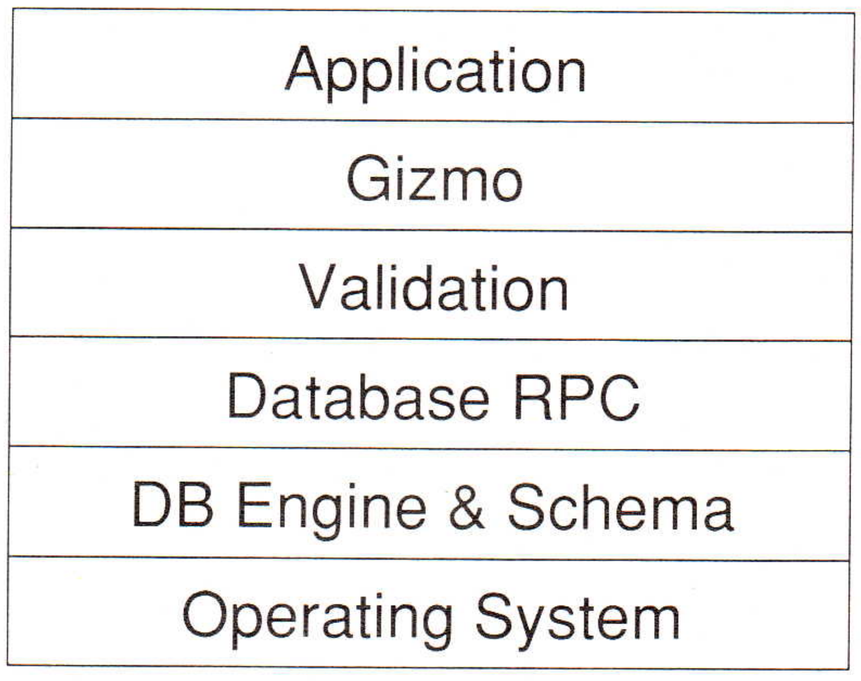

Figure 3: System Layers

continued on next page

Policy-Routing Configuration System *(continued)*

• *Operating System:* There is not much to be said about this layer. We currently run the database and server on RS6000's running AIX 3.2.

• *Database Engine and Schema:* This layer provides the DBMS for all the configuration data. The database models different flavors of networks, autonomous systems, routers, links, interfaces and routing sessions. It is important to note that we model both the internal configuration of the backbone and its external connections. We currently use Informix as our database engine but the independence of this level from those above it allows the use of nearly any DBMS in its place.

• *DB Remote Procedure Call:* This layer permits Config Database applications and the DBMS to run on different physical machines. We have developed a database RPC that allows a client process anywhere on the Internet to connect to the database server and issue database commands. A moderate level of security has been built into this RPC by using trusted hosts and users, similar to the UNIX r* commands: *rlogin, rsh,* etc. The database provides another level of security by restricting access down to individual columns.

• *Validation:* This layer validates all data before it is actually added to the database, enforcing constraints which the database engine cannot. These are generally application-specific constraints such as "a link cannot have the same interface at both ends" or "an EGP session must occur only between peer and backbone interfaces." These constraints are actually implemented as C routines and can therefore be quite complex.

• *Gizmo:* This layer makes the database applications independent from the underlying schema and query language by providing a mapping from high-level objects, as seen by the application, to actual tables and columns in the database. Gizmos come in several types, roughly one type for every entity being modeled, have attributes and are related to each other in a hierarchical fashion. Because an application only refers to gizmos and their attributes, this layer provides a certain degree of freedom in rearranging the schema and renaming columns and tables without breaking the applications.

• *Application:* This layer provides the user's interface to the database. Users interact with the database using clients like the NOC clients or the Configuration Operator's User Interface.

Futures

We are considering many exciting extensions for the system I've just described. Perhaps most important are those which will allow for the distributed input and output of data.

One can imagine replacing the e-mail NACR mentioned above with an X Windows client running on a peer network representative's workstation. Such a client might connect to a config changes database and allow them to submit their changes using a graphical interface. These changes could then be reviewed, processed and eventually batched into the real database. Submitting changes this way would have the advantages of giving the requester immediate feedback concerning the validity of certain aspects of his request and providing easy access to the status of earlier requests. For instance, the requester might see that the completion of a network addition placed last week is pending the approval of peer network B. This could be an effective way to scale the system as the number of network requests grows.

The dissemination of routing information is an important service that Merit currently provides by making configuration files and reports available for anonymous FTP. As mentioned above, some peer networks parse our configuration reports and generate configuration files for their own routers. This service could be enhanced by distributing routing clients that would run on a remote user's workstation and would query the configuration database for up to date routing information. The process could be automated even further by having the peer router periodically connect to the configuration database, read the necessary routing information and automatically update its own configuration.

Acknowledgements

Many people have made important contributions to this project but in the interest of brevity I'll only mention a few of them. The people actively working on this project are Dale Johnson, Chinh Nguyen, Steve Richardson, Rick Riolo, and myself. John Vollbrecht and Tom Libert were also largely responsible for pushing things along. Finally, Sue Hares should be commended for the early and forward-looking work she did on the Configuration System.

References

[1] Hares, S., "Components of OSI: Inter-Domain Routing Protocol (IDRP)," *ConneXions,* Volume 6, No. 5, May 1992.

[2] Hares, S. & Katz, D., "Administrative Domains and Routing Domains: A Model for Routing in the Internet," RFC 1136.

[3] *ConneXions,* Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[4] *ConneXions,* Volume 5, No. 1, January 1991, "Special Issue: Inter-domain Routing."

[5] Deborah Estrin, Yakov Rekhter, and Steve Hotz, "Scalable Inter-Domain Routing Architecture," In Proc. ACM SIGCOMM, 1992.

[6] Deborah Estrin, Yakov Rekhter, and Steve Hotz, "A Unified Approach to Inter-Domain Routing," RFC 1322, 1992.

[7] Kirk Lougheed and Yakov Rekhter, "A Border Gateway Protocol 3 (BGP-3)," RFC 1267, 1991.

[8] "Protocol for Exchange of Inter-domain Routeing Information among Intermediate Systems to Support Forwarding of ISO 8473 PDUs," ISO/IEC/ JTC1/SC6 DIS10747.

[9] Yakov Rekhter and Dave Katz, "The Border Gateway Protocol," *ConneXions,* Volume 5, No. 1, January 1991.

ANDY ADAMS holds a B.S. in Computer Engineering from the University of Michigan and has been involved with the NSFNET since 1988 when he joined Merit Network Inc. In addition to networking and databases, his interests include computer modeling and simulation and data visualization. E-mail: `ala@merit.edu`.

Traffic Shaping Using Leaky Bucket

by Craig Partridge, BBN Systems and Technologies

Introduction

As several researchers have pointed out, transmitting and receiving data at gigabit speeds is not the hardest problem related to the design of high speed networks. The really difficult problem appears to be achieving gigabit speeds while also providing service guarantees. These service guarantees include features like bounding the maximum delay or ensuring a video transmission will get enough bandwidth.

Traffic shaping

In this article, we'll look briefly at part of the problem of providing service guarantees: *traffic shaping*. The basic idea behind traffic shaping is the following. Assume we have a flow (or stream, or connection, or whatever is your favorite word for transmission path between two transport endpoints) which needs the network to guarantee that the end-to-end delay over the flow will be less than a certain amount, and the loss rate will not exceed a certain level. To make those kinds of guarantees, the network needs to know something about how the flow is going to behave. For example, if the flow is going to be sending 1 megabit packets at an average of 1 gigabit per second, a different path may be required than for a flow sending 1 Kbit packets at 56Kbps. The idea behind traffic shaping is to find ways to describe how the flow will place packets into the network (size of packets, average bandwidth, etc.) in such a way that the network both knows what to expect (and can punish flows which exceed the behavior promised), and can use the description of the flow to do scheduling inside the network.

An aside: if the network does no scheduling, it can clearly make no promises unless it is kept so underloaded that achieving all guarantees is trivial. If we do not wish to underload our networks, then we have to do some kind of traffic scheduling in at least some of the routers in the network. Exactly how much scheduling is required, however, is a hot topic of debate and how to implement it is a hot topic of debate. What we do know is that if we shape traffic at the edges of the network and do some sort of queueing in the middle, we can give performance guarantees. (See [1], [2], and [3].) But which form of queueing is to be preferred is unclear.

Probably the most popular way to do traffic shaping is by using a set of schemes collectively known as "leaky bucket." There are actually a number of variations on leaky bucket. We'll consider the three major versions here.

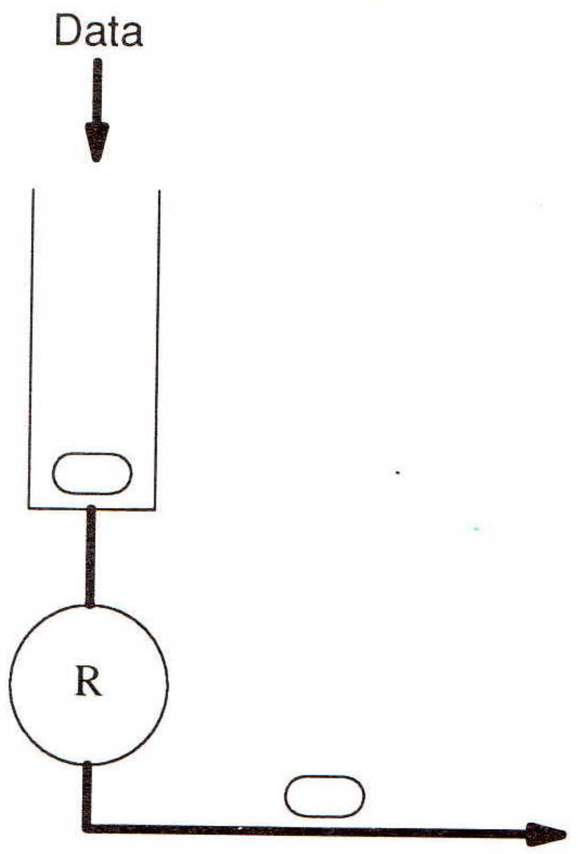

Figure 1: Simple Leaky Bucket

Leaky Bucket

The first and original version of leaky bucket is the simplest. It was designed to work with *Asynchronous Transfer Mode* (ATM). ATM transmits fixed-size packets called *cells*. Whenever a flow wants to send a bunch of cells, it puts the cells into a fixed-size queue, called a *bucket*. At some rate, *R*, the cells are removed, one by one, from the bottom of the bucket and sent over the network. If the flow tries to put too many cells into the bucket, it overflows. Essentially what simple leaky bucket does is convert a possibly bursty stream of cells from the flow, into a constant bit-rate transmission pattern on the network. A diagram of simple leaky bucket is shown in Figure 1.

The problem with simple leaky bucket is that it is too simple. A lot of data communications traffic has the property that its average bitrate is pretty low, but it has occasional large bursts of traffic. Simple leaky bucket will force the flow to describe itself as sending at the burst rate, rather than the average rate, if the flow wants to get enough bandwidth to send its bursts quickly. But since the flow will usually not send at the burst rate, the network will probably over-allocate resources to the flow. So traffic shaping according to simple leaky bucket is likely to lead to poor use of the network. For this reason, many researchers currently prefer another version of leaky bucket, sometimes called "token bucket."

Token Bucket

In a token bucket scheme, instead of holding data, the bucket holds tokens, where each token represents the right to send a certain amount of data (usually a byte or a cell). The bucket fills with credits at rate *R*. Whenever the bucket gets full, the extra credits "overflow" the bucket and are discarded. A diagram of token bucket is shown in Figure 2.

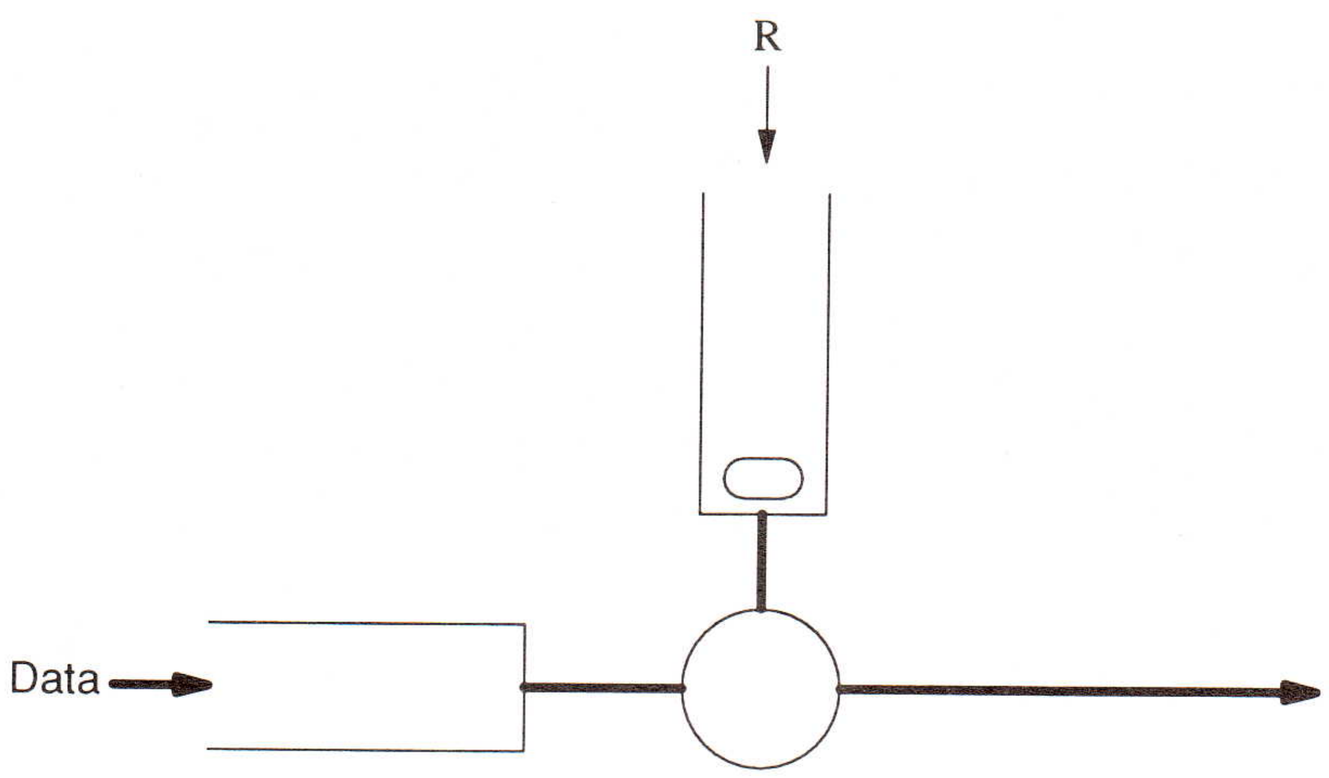

Figure 2: Token Bucket

When a flow wants to send some data (a packet or a cell), it looks in the token bucket to see if the bucket contains a number of tokens equal to the amount of data the flow wants to send. If there are enough tokens, the data is sent, and the tokens are removed from the bucket. If there are not enough tokens, the flow must wait until enough tokens have accumulated.

What token bucket does is allow a flow to be bursty (because if the bucket is full, the flow can send an amount of data equal to the size of the bucket without waiting) but according to a limit (after the bucket has emptied, the flow has to wait for it to start to fill again). Expressed formally, the token bucket scheme says that if the bucket size is *B*, in any time interval *T*, the maximum amount of data that the flow can send is equal to $B+(R*T)$ tokens.

continued on next page

Traffic Shaping Using Leaky Bucket *(continued)*

To see how we might use token bucket, consider a flow which occasionally (say an average of once a minute) sends a burst of 1 Mbit of data, but the rest of the time sends less data, an average of 10 Kbits a second. Leaky bucket would require a rate, *R*, of close to 1Mbps to keep up with the burst. With token bucket, one might specify a bucket that holds 1 Mbit of tokens, and a rate of about 30Kbps (which is enough to sustain 10Kbps throughput and completely refill the bucket every 50 seconds to deal with the bursts).

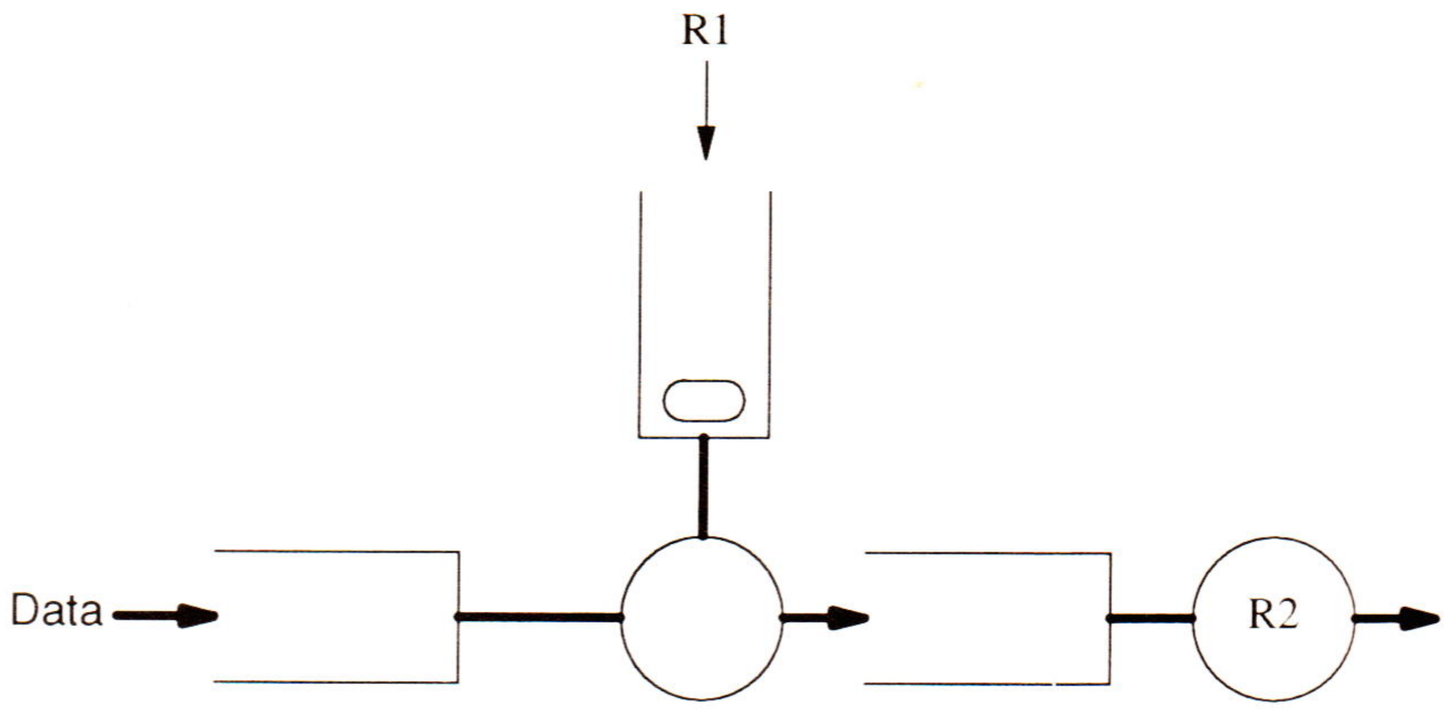

Figure 3: Token Bucket with Leaky Bucket

Variation There's one small problem with token bucket. What if the bucket size is really big? Then, in the worst case, a flow could blast a whole bucket's worth of data onto its network link without waiting. Now what if that network link is a LAN shared among several hosts? Perhaps we don't want one flow (and thus one host) to be able to hog the link for very long. One way to solve this problem is use both a token bucket and a leaky bucket in sequence. After the token bucket lets the data past, the data is thrown into the leaky bucket. The leaky bucket rate is set very high (much higher than the token bucket rate) but still below the link bandwidth. This way, even if the flow is blasting away, the leaky bucket ensures there is some bandwidth left for other hosts on the LAN. A diagram of this variation of token bucket is shown in Figure 3.

References

[1] A. K. J. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks (Ph.D. Thesis)," MIT Laboratory for Information and Decision Systems, Report LIDS-TH-2089, February 1992.

[2] S. J. Golestani, "A Stop-and-Go Queueing Framework for Congestion Management," in Proceedings ACM SIGCOMM '90, pp 8–18.

[3] D. Verma, H. Zhang, and D. Ferrari, "Guaranteeing Delay Jitter Bounds in Packet Switching Networks," Proceedings of TriComm '91, April 1991. (Available from the IEEE).

This article is an expanded version of a column that appeared in the ISOC Newsletter, Vol. 1, No. 3, Summer 1992.

CRAIG PARTRIDGE is a research scientist at BBN Systems and Technologies where he does research on gigabit networks. He teaches part-time at Stanford University and holds a Ph.D. from Harvard. He is also Editor-in-chief of *IEEE Network Magazine* and past editor of ACM SIGCOMM *Computer Communication Review.* Craig's book, *Gigabit Networking,* from Addison-Wesley is currently scheduled to be published in July of 1993. E-mail: craig@aland.bbn.com.

Letters to the Editor

Ole,

Regarding the October 1992 issue, you asked for OSI comments: I'm a former OSI believer, trying not to become a TCP/IP fanatic. However! Others have articulated the flaws in the OSI process very clearly. I believe that these issues of procedure are the most critical problems with OSI.

Then and now

We should also note that the key technical work on the Network and Transport Layers was done in the 1980–1984 timeframe, when X.25 was hot, and it *was* natural to follow the X.25 model. Do the solutions framed in 1983 make sense for 1993? Maybe for some, maybe not for others.

A lot has changed since then—and continues to change. Although OSI has been a fertile source of good ideas, I hope that we will solve the technical problems of the 90s by the usual Internet method of looking at the best technical solutions available today.

Naming and addressing

The most surprising failure and flaw in the CCITT/OSI efforts has been in addressing and naming. Their wide-open "you-choose-your-own-format" approach has made implementation very difficult. Considering that the CCITT has so often been successful in designing global numbering schemes, the fact that they just specified an *Authority Format Indicator* and said "roll your own" at layer three—and provided very little concrete guidance at higher layers—is surprising.

Of course, if you have an X.25 model in mind, finding a destination is performed only once, at the start of a connection. An overhead of a few seconds to figure out where you are going may be acceptable. If you are performing dynamic routing, the situation is very different.

Has OSI come and gone?

Perhaps the time for OSI has come and gone. Access to OSI literature has been effectively closed, since it is so costly to obtain. This has had an important consequence. The level of OSI knowledge, even among vendor field support, is low to non-existent. If the U.S. government were really serious about OSI, they would have realized that literacy in the subject was important, that they should be subsidizing the spread of information, and long ago should have campaigned vigorously in ISO and CCITT for free access to the materials. Note that having seen a picture of the seven-layer OSI stack does not comprise OSI literacy!

—*Sidnie Feit, The Standish Group*

For further reading

[1] Jacobsen, Ole, "The Trouble with OSI," *ConneXions,* Volume 6, No. 5, May 1992, p 62.

[2] desJardins, Richard, "*Opinion:* OSI is (Still) a Good Idea," *ConneXions,* Volume 6, No. 6, June 1992, p 33.

[3] Metcalfe, Bob, Smart, Bob, and Blackshaw, Bob, "Letters to the Editor," *ConneXions,* Volume 6, No. 6, June 1992, p 37.

[4] Rose, Marshall, "Comments on: '*Opinion:* OSI is (Still) a Good Idea,'" *ConneXions,* Volume 6, No. 8, August 1992, p 20.

[5] desJardins, Richard, "Comments on:' Comments on: *Opinion:* OSI is (Still) a Good Idea,' " *ConneXions,* Volume 6, No. 10, October 1992, p 43.

[6] desJardins, Richard, "Internet 2000," *ConneXions,* Volume 6, No. 10, October 1992, p 24.

[7] Hoffmann, Harald, "Letter to the Editor," *ConneXions,* Volume 6, No. 11, November 1992, p 31.

continued on next page

Letters to the Editor *(continued)*

Dear Ole,

Good intentions

The October 1992 *ConneXions* article, "Internet 2000" by Dick desJardins, had an enlightened desire to pursue a path of peace and integration, trying to resolve the long-standing feud between the TCP/IP and OSI camps. Unfortunately, good intentions are not enough. It would help, for example, if Dick's proposals responded to real and serious need, rather than emphasizing the occasional juvenile outbursts from members of one camp or the other. Worse, his suggestions often rely on incorrect technical information.

IPAE confusion

Dick's proposal referenced the current Internet effort to replace IP version 4 (IPv4). He rejected the *IP Address Encapsulation* (IPAE) proposal, of which I am an author, because he confused its technical details with an earlier proposal by Bob Hinden, the other author of IPAE. To set the record straight: IPAE uses standard, hierarchical, global addresses, just like the current IPv4 but with more bits. I hope that my survey article, in the *ConneXions* that followed Dick's article, will settle that matter. ["The ROAD to a New IP," November 1992.] For completeness, please note that IPAE is now positioned strictly as a transition technology, designed to permit an easy segue from IPv4 to *SIP,* the proposal by Steve Deering that also was mentioned in my article. (Its address format also is global and hierarchical, just like IPv4 and CLNP.)

But back to Internet 2000.

(One is tempted to ask if that is the date by which OSI will *finally* be ready, but I promised Dick not to get catty in this note...)

Conversion is difficult

Dick refers to an "anything over anything" approach to the integration of TCP/IP and OSI. This sounds appealing. What it ignores is the amount of work required to do such a range of mappings. Dick proposes a model of parallel operations, with cross-overs at various places in the 7-, 5-, or 4-layer stack (depending upon how one counts). Unfortunately, this completely avoids dealing with the essential requirement for interoperation. For example, X.400 works just fine over TCP/IP, as does SMTP/RFC 822. But users of each need to talk with each other and this requires conversion between the two. This is extremely difficult to accomplish on a global, stable and comfortable level. While e-mail gateways exist today and are essential, they are a continuing source of pain.

Coexistence

Worse, such conversions do not constitute integration. They constitute *coexistence.* Coexistence is useful but awkward. I don't believe it helps anyone to sugar-coat the situation by declaring coexistence as the means of making the two stacks all one happy family. Multiple stacks are a fact of life, and fighting over that fact isn't helpful. Neither is it helpful to foster more duplication than absolutely necessary.

The two stacks represent two different ways of doing business. When one community's stack lacks a facility adequately provided by the other, I hope that the former community uses the existing solution, rather than spending effort creating a new one. As Dick observed, the TCP/IP community didn't define any media-layer standards (until PPP was developed, quite recently) since other groups covered that ground. And the Internet has been a happy playground for X.500 experiments. There even is strong interest in various OSI routing and host-configuration work (IDRP, IS–IS and ES–IS).

Come to think of it, once one gets past the theatrical, public exchanges between the camps, the Internet has a reasonable track record of considering and using OSI protocols. I can't say that I've noticed the favor returned, though, with the possible exception of BGP, and even that is being modified substantially by the OSI camp, rather than being adopted directly. Even so, I suspect the Internet will make use of it.

But back to Internet 2000.

Using CLNP

Among those pursuing CLNP (TUBA) for use as the IPv4 replacement a common theme is the assertion that it is already installed in the Internet. Hence, we are told, conversion costs will be minimal. On mentioning this to an operations and support friend at a large, nearby university, I was informed that CLNP might be running in routers, but none of his 10,000 hosts was yet running CLNP. And none are scheduled to be. Perhaps the cost of converting to CLNP will be a bit larger than we are being told? Well, the article's title gives us 7 years to accomplish the conversion, but I'm not sure if that will be enough.

Most of the work that Dick suggests doing for *ping, traceroute,* the *Domain Name System* (DNS) and other software that needs to handle larger CLNP addresses must be done for *any* IP replacement. Nothing he mentions in anyway is better (or worse, I believe) if done for CLNP.

Regarding Dick's reference to RFC 1006, which permits OSI upper layers to operate over TCP, he apparently was unaware that RFC 1006 is now an Internet standard. However, it does suggest again that the OSI community could improve its acceptance of Internet technology. They might even try making RFC 1006 an International Standard...

Transport bridges

Dick then goes on to suggest a variety of activities that would result in being able to have hosts choose whatever transport they want, with powerful transport switches (transport bridges) fixing up the differences. Sorry, but transport bridges only work when the application layer is the *same* at both ends of the wire. In other words, this problem is much, much harder than Dick realizes and requires technology designed to facilitate *transition,* not *permanent* interoperation.

Underneath Dick's suggestion is the implication that there is a problem with TCP. It's almost 20 years old and often criticized as being archaic and in need of replacement. Never mind that every increase in media speed has resulted in improvements to TCP implementations (and sometimes to the protocol) which permit full utilization of the medium, even approaching gigabit speeds. In spite of a few wrinkles, TCP turns out to operate quite well in the modern Internet. Since there is no strong evidence that TCP is seriously broken, I suggest that we stop pursuing its replacement

Lack of cooperation

Dick suggests that we not "perpetuate the 'two-community' divisiveness into the next century" and I certainly concur with the desire. The only problem is that we have no history of successful cooperation between the formal Internet standards process and that of ISO and CCITT. We *do* have some examples of each group borrowing from the other. And we have examples of individuals working in both groups. But we have no examples of organizational inter-operation. While it would be a fine thing to see, I'd strongly suggest against anyone relying on it in their product or operations plans.

—Dave Crocker
(dcrocker@mordor.stanford.edu)
The Branch Office
On a chilly California day

Book Review

Guide to IT Standards Makers and Their Standards by Peter Judge, Technology Appraisals, 1991 (ISBN 1-871-80217-2). Our industry has been groping toward a mechanism for producing open technology for the last dozen years. Initially, most people's hopes were pinned on the official international standards bodies. As many of these groups' results are increasingly viewed as failures, vendor consortia and other more informal groups have stepped in to fill the gap. This book attempts to provide a survey of the currently active organizations producing open technology, officially and otherwise.

Wide coverage

Not too long ago, a book with this title would have been quite thin, describing only the official bodies. It's an indication of the frenzy for open systems that the book runs well over 200 pages and covers almost 30 different organizations. Beginning with the traditional favorites—ISO and CCITT—it goes on to describe some of their offshoots, groups like the *European Workshop for Open Systems* and the (since renamed) NIST *OSI Implementors Workshop*. The book then provides descriptions of a wide range of industry and professional bodies, including the IEEE, the *Internet Architecture Board* (IAB), the *Object Management Group, X/Open,* the *Open Software Foundation,* and more. The final section describes a potpourri of user-oriented groups. For each organization, the author describes its mission, structure, history, relation to other groups, and deliverables.

Coherent

Overall, this is an excellent book. It provides the most complete and thorough coverage I have seen of a very confusing but undeniably important part of the world. The author has done a good job of coalescing a large amount of information into a coherent, readable book. Having said that, the book is not without flaws. My expectation in reviewing a book about standards organizations, especially one written by a European and using the popular-mostly-in-Europe acronym "IT" in its title, was to find a heavy tilt toward OSI and its creators. I was not disappointed. To be fair, any book attempting to describe all standards bodies must devote a substantial chunk of its space to those working on OSI, if only because there are so many of them. All of the players are on stage here, and all are described with a straight face.

OSI bias

While this complete coverage of the environment is commendable, the author's personal tilt toward OSI leads him astray at times. No mention is made of the increasingly obvious fact that a majority of the technology produced by this surfeit of groups will never actually be used. Also, the coverage of non-OSI groups usually manages to include a section on how they will migrate toward, interwork with, or be replaced by OSI, while the converse is not true. In the section on the IAB, for example, the author makes several references to the "official US policy" of OSI and of the need to accommodate OSI in the Internet (he also repeats the widely disseminated myth that OSI applications like VT and FTAM have more functionality than their TCP/IP counterparts—I wish him luck trying to find users of both who agree). It's becoming increasingly obvious to those of us who live here that, official policy or not, TCP/IP is here to stay, and OSI's chances of playing anything more than a marginal role are shrinking daily.

Useful book

While the reader is cautioned to filter out some of the author's bias, this book nevertheless provides an excellent guide to difficult terrain. Maybe one day all the parties will agree on a single process to create vendor-neutral technology, and maybe a future edition of this book will be much thinner, describing only a single organization. Until then, anyone working in this area will find this book useful.

—David Chappell, Chappell & Associates

Call for Papers

The USENIX *Symposium on Mobile & Location-Independent Computing* will be held in Cambridge, Massachusetts, August 2–3, 1993.

Much of the growth of UNIX has been due to its support for casual communications, thus fostering cooperative work within a location-independent framework. The latest incarnation of location independence is Mobile Computing. Distributed computing, now fashionable in other circles, was pioneered by the UNIX community. Support for Mobile Computing is the next logical step in assuring the role of UNIX as the operating system that offers a rich and complete feature set.

Topics

Progress in Mobile Computing is everywhere evident both in academic and non-academic circles. We intend to concentrate on it in a true state-of-the-art symposium and technical free-for-all on what it takes to make Mobile Computing work and work right. The workshop will address many issues and ongoing developments, including, but not limited to:

- Naming (e.g., *Prospero* or OSF/DCE DNS)
- Wide area information distribution (e.g., WAIS and *archie*)
- Security (e.g., authentication based on devices and digital signature services)
- User locatability (e.g., paging systems and active badges)
- Rendezvous (e.g., videoconferencing over the Internet and various Groupware efforts)
- Networking and Connectability (e.g., the new IETF routing work, movement of "sockets" from site to site, and the rumored advent of IP connections from airplanes)
- Portable tiny devices (e.g., the various palmtops and personal information assistants)

As is usual for a USENIX symposium, we are looking for new and arresting developments in systems that directly contribute to a technical understanding of Mobile Computing. UNIX will be the lingua franca of discussion, but we are eager for progress from other world views to be presented as well. This symposium will have limited attendance.

Submissions

Extended abstracts of 1500–2500 words (9000–15000 bytes or 3–5 pages) should be sent to Dan Geer at the address below (those submitting hardcopy abstracts must send five copies). Shorter abstracts run a significant risk of rejection as there will be little on which the program committee can base an opinion.

Important dates

April 19, 1993	Extended abstracts due
May 3, 1993	Notification to authors
June 14,1993	Camera-ready copy due

More information

For further information about the symposium, contact:

Daniel E. Geer, Program Chair
Geer Zolot Associates
200 Portland Street
Boston, MA 02114
E-mail: geer@world.std.com
Phone: +1 617-367-2010 • Fax: +1 617-367-6131

Announcement and Call for Papers

The *USENIX Summer 1993 Technical Conference* will be held in Cincinnati, Ohio, June 21–25, 1993.

Theme

A little over ten years ago UNIX encountered the bitmap display and the mouse. Developments since then, such as The X Window System, didn't try to change UNIX. Rather they layered on it to cope with the demands of the new user interface technology. After ten years this doesn't look like a successful strategy; UNIX hardly has industry-leading user interfaces and a horde of new user interface technologies are arriving.

Radical thinking and new operating system capabilities are needed to support new user interface technologies. Communicating with the user is a real-time problem, why aren't we using the emerging real-time capabilities of UNIX to support it? Are UNIX byte-string files adequate, or do we need a generalized file attribute model? Can users really navigate a file name space that is a rooted tree of all the files in the Internet?

As usual, USENIX is interested in papers describing new and interesting developments in open operating systems. But in Cincinnati we're particularly interested in papers addressing the evolution of operating systems to support new and effective user interfaces.

Keynote speaker

Our keynote speaker, Bruce Tognazzini, has been a long-time customer of the operating system support for the user interface. He has been designing man-machine interfaces for better than 30 years. He spent the last 14 years at Apple where he led at various times both the Apple II and Macintosh human interface efforts before moving to SunSoft earlier this year. During his most recent tenure in the Evangelism group at Apple, he wrote what the author and his mom have both described as a major new publication in the field of human-computer interaction, "Tog on Interface."

Important dates

Abstracts Due:	February 2, 1993
Notifications to Authors:	February 27, 1993
Camera-Ready Copy Due:	April 14, 1993

Work-in-progress Sessions

Work-in-Progress (WIP) sessions provide attendees with an opportunity to present short (typically 10 minute) talks on work which is on-going, currently under development or is newly completed. This year USENIX is particularly encouraging students to present their research at the WIP sessions. The USENIX audience provides valuable discussion and feedback. The WIP coordinator is Peg Schafer, BBN, Phone: 617-873-2626, E-mail: `wip@usenix.org`. Time for a talk may be reserved in advance by phone or e-mail, or on-site.

Submissions

Authors of papers to be presented at the technical sessions and published in the proceedings must submit one copy of an extended abstract to:

Summer 93 USENIX
USENIX Association
2560 Ninth St, Suite 215
Berkeley, CA 94710
E-mail: `summer93papers@usenix.org`
Fax : 510-548-5738

Enquiries about these submissions may be made by e-mail to `david@usenix.org` or to 510-528-8649. Please use at least *two* of the above submission methods (mail, e-mail, fax).

The schedule for reviewing submissions for the conference is very short, and reviewers don't have time to read full papers. The object of an extended abstract is to convince the reviewers that a good paper and 25-minute presentation will result. They need to know that the authors:

- Are attacking a significant problem.
- Are familiar with the current literature about the problem.
- Have devised an original solution.
- Have implemented it and, if appropriate, characterized its performance.
- Have drawn appropriate conclusions about what they have learned and why it is important.

As at previous USENIX conferences, papers that analyze problem areas and draw important conclusions from practical experience are welcome. Note that the USENIX conference, like most conferences and journals, considers it unethical to submit the same paper simultaneously to more than one conference or publication, or to submit a paper that has been or will be published elsewhere.

The extended abstract must be 5 sides or less. Only the first 5 sides of your submission will be sent to the reviewers. The full paper may be attached to the extended abstract; it will not be sent to the reviewers but may be helpful during final evaluation.

The extended abstract should include the abstract as it will appear in the final paper, and represent the paper in "short form." Supporting material may be in note form. Authors should include references to establish that they are familiar with the literature, and if appropriate performance data to establish that they have a working implementation and measurement tools.

Every submission should include one additional side containing:

- The name, surface mail address, daytime and evening phone numbers, E-mail address and (if available) fax number of one of the authors, who will act as the contact point.
- An indication of which, if any, of the authors are students
- A list of audio/visual equipment desired beyond a microphone and an overhead projector.

Authors of accepted submissions will be notified by February 27, 1993. They will receive instructions for preparing camera-ready copy of an 8–12 page final paper, which must be received by March 14, 1993.

Program Committee

David S. H. Rosenthal	SunSoft Inc (Program Chair)
Matt Blaze	AT&T Bell Laboratories
Nathaniel Borenstein	Bellcore
Bob Gray	US West Advanced Technologies
Steve Kleiman	SunSoft Inc.
Kirk McKusick	University of California at Berkeley
Jeff Mogul	Digital Equipment Corp.
JR Oldroyd	Instruction Set
Pat Parseghian	AT&T Bell Laboratories
Dennis Ritchie	AT&T Bell Laboratories

Announcement and Call for Papers

The Australian UNIX Users' Group (AUUG) Inc., Forum for Open Systems Users will host *AUUG '93* in Darling Harbour, Sydney, Australia, September 27–30, 1993.

Theme

Over the past several years we have heard about "What are Open Systems," and "Maintaining Control with Open Systems." Now it's time to hear about the results which have been achieved. Rapid expansion, the challenge of integration, global networking, and security are all issues of importance and concern to users around the world. *AUUG '93* solicits papers on all aspects of UNIX and open systems, and particularly on successful applications and implementations of open systems technology to age-old and newly emerging problems. The first day will be devoted to tutorial presentations, followed by three days of papers, work-in-progress sessions and BOFs.

Tutorials

Provisions for two full-day tutorials and up to eight half-day tutorials have been made. These sessions, typically in a lecture format, are targeted to educate the audience and arm them with innovative "how to" lessons. Please submit tutorial abstracts, along with preference for a half- or full-day slot to the address below.

Papers

AUUG '93 provides dual Technical and Management tracks for the presentations. To share your innovative implementations, applications, and similar areas submit your abstract for the technical track. We are also interested in your experiences, case studies, strategic issues, and the like. If your topic better fits these areas submit your abstract for the Management track. The above should not, of course, discourage papers which are appropriate for both audiences at once. Vendor product announcements will be automatically rejected unless specifically submitted for the special advertising stream.

Best Student Paper Prize

A cash prize of $500 will be awarded for the best paper submitted by a full-time student at an accredited tertiary education institution. In addition, the ten "runners-up" will be rewarded with free registration.

Work-in-progress and advertising sessions

These brief 15 minute sessions are designed to report on current work with fundamental aspects highlighted. New to the AUUG conference are the Advertising sessions. These are devoted to new products only. Product specification sheets should be submitted with your abstract.

BOFs

Are you interested in discussing particular problem areas, sharing arcana on favourite programs, using the internet, or other controversial topics? During the lunch hour and at the end of each presentation day, one hour time slots for *Birds-of-a-Feather Sessions* (BOFs) will be available. We distinguish two types of BOF; *general interest* and *vendor sponsored*. Please contact the Programme Committee if you would like to organise a BOF. There may be some facilities charge to vendor sponsored events.

Submissions

Please indicate whether your submission is relevant to the technical or management audience, or both. In either case, submissions are required to be in the form of an abstract and an outline. Please provide sufficient detail to allow the committee to make a reasoned decision about the final paper; of course a full paper is also perfectly acceptable. A submission should be from 2–5 pages and include:

- Author name(s), postal addresses, telephone numbers, FAX and e-mail addresses.
- A biographical sketch not to exceed 100 words.
- Abstract: 100 words.

- Outline: 1–4 pages giving details of the approach or algorithms pursued. Shorter outlines will not give the programme committee enough information to judge your work fairly, and, in most cases, this means your paper will be rejected. Longer outlines and full papers simply cannot be read by the committee in the time available. However, you may append a full paper to your outline; this is sometimes useful during evaluation.
- References to any relevant literature.
- Audio-visual requirements: 35 mm slides are preferred, however, overheads will be accepted. Hand written or typewriter generated overheads will not be accepted.

Please submit one hard copy and one electronic copy (if possible) to the addresses below:

AUUG '93 Programme
P.O. Box 366
Kensington, New South Wales 2033
AUSTRALIA
Phone: +61 2 361-5994 • Fax: +61 2 332-4066
E-mail: auug93@cs.su.oz.au

Important dates

Abstract and outlines due:	April 6, 1993
Notifications to authors:	April 26, 1993
Final Papers due:	July 26, 1993

Call for Papers

About the journal

The *Asia-Pacific Engineering Journal* (APEJ) provides within the rapidly changing Asia-Pacific region a unique source of information on current international research activities and trends in technology. It aims to keep its readers fully briefed with major papers, reports and reviews on state-of-the-art technologies and products. The journal is published in separate parts that cover the disciplines of Electrical, Mechanical, Civil, Chemical and Industrial Engineering. Part A is devoted to Electrical Engineering, and covers the four main areas of Communications Engineering, Computer Engineering, Control and Automation, and Microelectronics.

Topics

The December 1993 issue of Part A of the APEJ is devoted to Communications Engineering. The issue will be a special issue concentrating on the field of High-Speed Networking, and original contributions in all aspects of this field of research are now solicited. In particular, topics of interest include, but are not limited to, the following:

- Broadband ISDN and ATM networks
- Lightwave networks
- Flow and congestion control
- Gigabit/s networking
- Transport protocols
- Management

Submissions

Prospective authors are requested to submit four (4) copies of their manuscripts, written in English, and including a 100-word abstract, to the following by 1 April 1993:

Dr. Kee C. Chua
Department of Electrical Engineering
National University of Singapore
10 Kent Ridge Crescent
SINGAPORE 0511
Fax.: +65 777-3117
E-mail.: eleckc@nuscc.nus.sg

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

ADDRESS CORRECTION
REQUESTED

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1